

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: June 21, 2019

R. Bush
IIJ Lab / Dragon Research Lab
S. Turner
sn3rd
K. Patel
Arrcus, Inc.
December 18, 2018

Router Keying for BGPsec
draft-ietf-sidrops-rtr-keying-02

Abstract

BGPsec-speaking routers are provisioned with private keys in order to sign BGPsec announcements. The corresponding public keys are published in the global Resource Public Key Infrastructure, enabling verification of BGPsec messages. This document describes two methods of generating the public-private key-pairs: router-driven and operator-driven.

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [BCP 14](#) [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on January 16, 2017.

Copyright Notice

Copyright (c) 2018 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1.	Introduction	2
2.	Management / Router Communication	3
3.	Exchange Certificates	4
4.	Set-Up	4
5.	Generate PKCS#10	4
5.1.	Router-Generated Keys	5
5.2.	Operator-Generated Keys	5
5.2.1.	Using PKCS#8 to Transfer Private Key	5
6.	Send PKCS#10 and Receive PKCS#7	6
7.	Install Certificate	6
8.	Advanced Deployment Scenarios	7
9.	Key Management	8
9.1.	Key Validity	8
9.2.	Key Roll-Over	9
9.3.	Key Revocation	9
9.4.	Router Replacement	10
10.	Security Considerations	10
11.	IANA Considerations	12
12.	References	12
12.1.	Normative References	12
12.1.	Informative References	13
Appendix A.	Management/Router Channel Security	14
Appendix B.	The n00b Guide to BGPsec Key Management	15
	Authors' Addresses	18

[1.](#) Introduction

BGPsec-speaking routers are provisioned with private keys, which allow them to digitally sign BGPsec announcements. To verify the signature, the public key, in the form of a certificate [[RFC8209](#)], is

published in the Resource Public Key Infrastructure (RPKI). This document describes provisioning of BGPsec-speaking routers with the appropriate public-private key-pairs. There are two methods, router-driven and operator-driven.

These two methods differ in where the keys are generated: on the router in the router-driven method, and elsewhere in the operator-driven method. Routers are required to support at least one of the methods in order to work in various deployment environments. Some routers may not allow the private key to be off-loaded while others may. While off-loading private keys would ease swapping of routing engines, exposure of private keys is a well known security risk.

In the operator-driven method, the operator generates the private/public key-pair and sends it to the router.

In the router-driven method, the router generates its own public/private key-pair.

The router-driven model mirrors the model used by traditional PKI subscribers; the private key never leaves trusted storage (e.g., Hardware Security Module). This is by design and supports classic PKI Certification Policies for (often human) subscribers which require the private key only ever be controlled by the subscriber to ensure that no one can impersonate the subscriber. For non-humans, this model does not always work. For example, when an operator wants to support hot-swappable routers, the same private key needs to be installed in the soon-to-be online router that was used by the the soon-to-be offline router. This motivated the operator-driven model.

Sections [2](#) through [7](#) describe the various steps involved for an operator to use the two methods to provision new and existing routers. The methods described involve the operator configuring the two end points (i.e., the management station and the router) and acting as the intermediary. [Section 8](#) describes another method that requires more capable routers.

Useful References: [\[RFC8205\]](#) describes gritty details, [\[RFC8209\]](#) specifies the format for the PKCS#10 certification request, and [\[RFC8208\]](#) specifies the algorithms used to generate the PKCS#10's signature.

2. Management / Router Communication

Operators are free to use either the router-driven or operator-driven method as supported by the platform. Regardless of the method chosen, operators first establish a protected channel between the

management system and the router. How this protected channel is established is router-specific and is beyond scope of this document. Though other configuration mechanisms might be used, e.g. NetConf (see [RFC6470]); for simplicity, in this document, the protected channel between the management platform and the router is assumed to be an SSH-protected CLI. See [Appendix A](#) for security considerations for this protected channel.

3. Exchange Certificates

A number of options exist for the operator management station to exchange PKI-related information with routers and with the RPKI including:

- Use application/pkcs10 media type [RFC5967] to extract certificate requests and application/pkcs7-mime [I-D.lamps-rfc5751-bis] to return the issued certificate,
- Use FTP or HTTP per [RFC2585], and
- Use Enrollment over Secure Transport (EST) protocol per [RFC7030].

4. Set-Up

To start, the operator uses the protected channel to install the appropriate RPKI Trust Anchor's Certificate (TA Cert) in the router. This will later enable the router to validate the router certificate returned in the PKCS#7 certs-only message [I-D.lamps-rfc5751-bis].

The operator also configures the Autonomous System (AS) number to be used in the generated router certificate. This may be the sole AS configured on the router, or an operator choice if the router is configured with multiple ASs. A router with multiple ASs can be configured with multiple router certificates by following the process of this document for each desired certificate.

The operator configures or extracts from the router the BGP Identifier [RFC4271] to be used in the generated router certificate. In the case where the operator has chosen not to use unique per-router certificates, a BGP Identifier of 0 MAY be used.

5. Generate PKCS#10

The private key, and hence the PKCS#10 certification request, which is sometimes referred to as a Certificate Signing Request (CSR), may be generated by the router or by the operator.

The PKCS#10 request SHOULD be saved to enable verifying that the

returned public key in the certificate corresponds to the private used to generate the signature on the CSR.

NOTE: The PKCS#10 certification request does not include the AS number or the BGP Identifier for the router certificate. Therefore, the operator transmits the AS it has chosen on the router and the BGP Identifier as well when it sends the CSR to the CA.

5.1. Router-Generated Keys

In the router-generated method, once the protected channel is established and the initial Set-Up ([Section 4](#)) performed, the operator issues a command or commands for the router to generate the public/private key pair, to generate the PKCS#10 certification request, and to sign the PKCS#10 certification request with the private key. Once generated, the PKCS#10 certification request is returned to the operator over the protected channel.

The operator includes the chosen AS number and the BGP Identifier when it sends the CSR to the CA.

NOTE: If a router were to communicate directly with a CA to have the CA certify the PKCS#10 certification request, there would be no way for the CA to authenticate the router. As the operator knows the authenticity of the router, the operator mediates the communication with the CA.

5.2. Operator-Generated Keys

In the operator-generated method, the operator generates the public/private key pair on a management station and installs the private key into the router over the protected channel. Beware that experience has shown that copy and paste from a management station to a router can be unreliable for long texts.

The operator then creates and signs the PKCS#10 certification request with the private key; the operator includes the chosen AS number and the BGP Identifier when it sends the CSR to the CA.

Even if the operator cannot extract the private key from the router, this signature still provides a linkage between a private key and a router. That is, the operator can verify the proof of possession (POP), as required by [[RFC6484](#)].

5.2.1. Using PKCS#8 to Transfer Private Key

A private key can be encapsulated in a PKCS#8 Asymmetric Key Package [[RFC5958](#)] and should be further encapsulated in Cryptographic Message

Syntax (CMS) SignedData [[RFC5652](#)] and signed with the operators's End Entity (EE) private key.

The router SHOULD verify the signature of the encapsulated PKCS#8 to ensure the returned private key did in fact come from the operator, but this requires that the operator also provision via the CLI or include in the SignedData the RPKI CA certificate and relevant operator's EE certificate(s). The router should inform the operator whether or not the signature validates to a trust anchor; this notification mechanism is out of scope.

6. Send PKCS#10 and Receive PKCS#7

The operator uses RPKI management tools to communicate with the global RPKI system to have the appropriate CA validate the PKCS#10 certification request, sign the key in the PKCS#10 (i.e., certify it) and generate a PKCS#7 certs-only message, as well as publishing the certificate in the Global RPKI. External network connectivity may be needed if the certificate is to be published in the Global RPKI.

After the CA certifies the key, it does two things:

1. Publishes the certificate in the Global RPKI. The CA must have connectivity to the relevant publication point, which in turn must have external network connectivity as it is part of the Global RPKI.
2. Returns the certificate to the operator's management station, packaged in a PKCS#7 certs-only message, using the corresponding method by which it received the certificate request. It SHOULD include the certificate chain below the TA Certificate so that the router can validate the router certificate.

In the operator-generated method, the operator SHOULD extract the certificate from the PKCS#7 certs-only message, and verify that the private key it holds corresponds to the returned public key. If the operator saved the PKCS#10 it can check this correspondence by comparing the public key in the CSR to the public key in the returned certificate. If the operator has not saved the PKCS#10, it can check this correspondence by generating a signature on any data and then verifying the signature using the returned certificate.

In the operator-generated method, the operator has already installed the private key in the router (see [Section 5.2](#)).

7. Install Certificate

The operator provisions the PKCS#7 certs-only message into the router

over the protected channel.

The router SHOULD extract the certificate from the PKCS#7 certs-only message and verify that the public key corresponds to the stored private key. If the router stored the PKCS#10, it can check this correspondence by comparing the public key in the CSR to the public key in the returned certificate. If the router did not store the PKCS#10, it can check this correspondence by generating a signature on any data and then verifying the signature using the returned certificate. The router SHOULD inform the operator whether it successfully received the certificate and whether or not the keys correspond; the mechanism is out of scope.

The router SHOULD also verify that the returned certificate validates back to the installed TA Certificate, i.e., the entire chain from the installed TA Certificate through subordinate CAs to the BGPsec certificate validate. To perform this verification, the CA certificate chain needs to be returned along with the router's certificate in the PKCS#7 certs-only message. The router SHOULD inform the operator whether or not the signature validates to a trust anchor; this notification mechanism is out of scope.

NOTE: The signature on the PKCS#8 and Certificate need not be made by the same entity. Signing the PKCS#8, permits more advanced configurations where the entity that generates the keys is not the direct CA.

8. Advanced Deployment Scenarios

More PKI-capable routers can take advantage of this increased functionality and lighten the operator's burden. Typically, these routers include either pre-installed manufacturer-generated certificates (e.g., IEEE 802.1 AR [[802.1AR](#)]) or pre-installed manufacturer-generated Pre-Shared Keys (PSK) as well as PKI-enrollment functionality and transport protocol, e.g., CMC's "Secure Transport" [[RFC7030](#)] or the original CMC transport protocol's [[RFC5273](#)]. When the operator first establishes a protected channel between the management system and the router, this pre-installed key material is used to authenticate the router.

The operator burden shifts here to include:

1. Securely communicating the router's authentication material to the CA prior to operator initiating the router's CSR. CAs use authentication material to determine whether the router is eligible to receive a certificate. Authentication material at a minimum includes the router's AS number and BGP Identifier as well as the router's key material, but can also include

additional information. Authentication material can be communicated to the CA (i.e., CSRs signed by this key material are issued certificates with this AS and BGP Identifier) or to the router (i.e., the operator uses the vendor-supplied management interface to include the AS number and BGP Identifier in the router-generated CSR).

2. Enabling the router to communicate with the CA. While the router-to-CA communications are operator-initiated, the operator's management interface need not be involved in the communications path. Enabling the router-to-CA connectivity MAY require connections to external networks (i.e., through firewalls, NATs, etc.).

Once configured, the operator can begin the process of enrolling the router. Because the router is communicating directly with the CA, there is no need for the operator to retrieve the PKCS#10 certification request from the router as in [Section 5](#) or return the PKCS#7 certs-only message to the router as in [Section 6](#). Note that the checks performed by the router in [Section 7](#), namely extracting the certificate from the PKCS#7 certs-only message, verifying the public key corresponds to the private key, and that the returned certificate validated back to an installed trust anchor, SHOULD be performed. Likewise, the router SHOULD notify the operator if any of these fail, but this notification mechanism is out of scope.

When a router is so configured, the communication with the CA SHOULD be automatically re-established by the router at future times to renew or rekey the certificate automatically when necessary (See [Section 9](#)). This further reduces the tasks required of the operator.

9. Key Management

Key management does not only include key generation, key provisioning, certificate issuance, and certificate distribution. It also includes assurance of key validity, key roll-over, and key preservation during router replacement. All of these responsibilities persist for as long as the operator wishes to operate the BGPsec-speaking router.

9.1. Key Validity

It is critical that a BGPsec speaking router is signing with a valid private key at all times. To this end, the operator needs to ensure the router always has a non-expired certificate. I.e. the key used to sign BGPsec announcements always has an associated certificate whose expiry time is after the current time.

Ensuring this is not terribly difficult but requires that either:

1. The router has a mechanism to notify the operator that the certificate has an impending expiration, and/or
2. The operator notes the expiry time of the certificate and uses a calendaring program to remind them of the expiry time, and/or
3. The RPKI CA warns the operator of pending expiration, and/or
4. The operator uses some other kind of automated process to search for and track the expiry times of router certificates.

It is advisable that expiration warnings happen well in advance of the actual expiry time.

Regardless of the technique used to track router certificate expiry times, it is advisable to notify additional operators in the same organization as the expiry time approaches, thereby ensuring that the forgetfulness of one operator does not affect the entire organization.

Depending on inter-operator relationship, it may be helpful to notify a peer operator that one or more of their certificates are about to expire.

9.2. Key Roll-Over

Routers that support multiple private keys also greatly increase the chance that routers can continuously speak BGPsec because the new private key and certificate can be obtained and distributed prior to expiration of the operational key. Obviously, the router needs to know when to start using the new key. Once the new key is being used, having the already distributed certificate ensures continuous operation.

More information on how to proceed with a Key Roll-Over is described in [[I-D.sidrops-bgpsec-rollover](#)].

9.3. Key Revocation

In certain circumstances, a router's BGPsec certificate may need to be revoked. When this occurs, the operator needs to use the RPKI CA system to revoke the certificate by placing the router's BGPsec certificate on the Certificate Revocation List (CRL) as well as re-keying the router's certificate.

When an active router key is to be revoked, the process of requesting

the CA to revoke, the process of the CA actually revoking the router's certificate, and then the process of re-keying/renewing the router's certificate, (possibly distributing a new key and certificate to the router), and distributing the status, takes time during which the operator must decide how they wish to maintain continuity of operations, with or without the compromised private key, or whether they wish to bring the router offline to address the compromise.

Keeping the router operational and BGPsec-speaking is the ideal goal; but, if operational practices do not allow this, then reconfiguring the router to disable BGPsec is likely preferred to bringing the router offline.

Routers which support more than one private key, where one is operational and other(s) are soon-to-be-operational, facilitate revocation events because the operator can configure the router to make a soon-to-be-operational key operational, request revocation of the compromised key, and then make a next generation soon-to-be-operational key. Hopefully, all this can be done without needing to take offline or reboot the router. For routers which support only one operational key, the operators should create or install the new private key, and then request revocation of the certificate corresponding to the compromised private key.

9.4. Router Replacement

Currently routers often generate private keys for uses such as SSH, and the private keys may not be seen or off-loaded from the router. While this is good security, it creates difficulties when a routing engine or whole router must be replaced in the field and all software which accesses the router must be updated with the new keys. Also, any network based initial contact with a new routing engine requires trust in the public key presented on first contact.

To allow operators to quickly replace routers without requiring update and distribution of the corresponding public keys in the RPKI, routers SHOULD allow the private BGPsec key to be inserted via a protected channel, e.g., SSH, NetConf (see [[RFC6470](#)]), SNMP. This lets the operator escrow the old private key via the mechanism used for operator-generated keys, see [Section 5.2](#), such that it can be re-inserted into a replacement router. The router MAY allow the private key to be to be off-loaded via the protected channel, but this SHOULD be paired with functionality that sets the key into a permanent non-exportable state to ensure that it is not off-loaded at a future time by unauthorized operations.

10. Security Considerations

The router's manual will describe whether the router supports one, the other, or both of the key generation options discussed in the earlier sections of this draft as well as other important security-related information (e.g., how to SSH to the router). After familiarizing one's self with the capabilities of the router, an operator is encouraged to ensure that the router is patched with the latest software updates available from the manufacturer.

This document defines no protocols. So, in some sense, it introduces no new security considerations. However, it relies on many others and the security considerations in the referenced documents should be consulted; notably, those document listed in [Section 1](#) should be consulted first. PKI-relying protocols, of which BGPsec is one, have many issues to consider - so many, in fact, entire books have been written to address them; so listing all PKI-related security considerations is neither useful nor helpful. Regardless, some bootstrapping-related issues are listed here that are worth repeating:

Public-Private key pair generation: Mistakes here are, for all, practical purposes catastrophic because PKIs rely on the pairing of a difficult to generate public-private key pair with a signer; all key pairs MUST be generated from a good source of non-deterministic random input [[RFC4086](#)].

Private key protection at rest: Mistakes here are, for all, practical purposes catastrophic because disclosure of the private key allows another entity to masquerade as (i.e., impersonate) the signer; all private keys MUST be protected when at rest in a secure fashion. Obviously, how each router protects private keys is implementation specific. Likewise, the local storage format for the private key is just that, a local matter.

Private key protection in transit: Mistakes here are, for all, practical purposes catastrophic because disclosure of the private key allows another entity to masquerade as (i.e., impersonate) the signer; transport security is therefore strongly RECOMMENDED. The level of security provided by the transport layer's security mechanism SHOULD be commensurate with the strength of the BGPsec key; there's no point in spending time and energy to generate an excellent public-private key pair and then transmit the private key in the clear or with a known-to-be-broken algorithm, as it just undermines trust that the private key has been kept private. Additionally, operators SHOULD ensure the transport security mechanism is up to date, in order to addresses all known implementation bugs.

Though the CA's certificate is installed on the router and used to verify that the returned certificate is in fact signed by the CA, the

revocation status of the CA's certificate is rarely checked as the router may not have global connectivity or CRL-aware software. The operator MUST ensure that the installed CA certificate is valid.

11. IANA Considerations

This document has no IANA Considerations.

12. References

12.1. Normative References

- [I-D.sidrops-bgpsec-rollover]
Weis, B, R. Gagliano, and K. Patel, "BGPsec Router Certificate Rollover", [draft-ietf-sidrops-bgpsec-rollover](#) (work in progress), December 2017.
- [I-D.lamps-rfc5751-bis]
Schaad, J., Ramsdell, B, S. Turner,
"Secure/Multipurpose Internet Mail Extension (S/MIME) Version 4.0", [draft-ietf-lamps-rfc5751-bis](#) (work in progress), July 2018.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC4086] Eastlake 3rd, D., Schiller, J., and S. Crocker,
"Randomness Requirements for Security", [BCP 106](#), [RFC 4086](#), DOI 10.17487/RFC4086, June 2005, <<https://www.rfc-editor.org/info/rfc4086>>.
- [RFC4253] Ylonen, T. and C. Lonvick, Ed., "The Secure Shell (SSH) Transport Layer Protocol", [RFC 4253](#), DOI 10.17487/RFC4253, January 2006, <<https://www.rfc-editor.org/info/rfc4253>>.
- [RFC4271] Rekhter, Y., Ed., Li, T., Ed., and S. Hares, Ed., "A Border Gateway Protocol 4 (BGP-4)", [RFC 4271](#), DOI 10.17487/RFC4271, January 2006, <<https://www.rfc-editor.org/info/rfc4271>>.
- [RFC5652] Housley, R., "Cryptographic Message Syntax (CMS)", STD 70, [RFC 5652](#), DOI 10.17487/RFC5652, September 2009, <<https://www.rfc-editor.org/info/rfc5652>>.
- [RFC5958] Turner, S., "Asymmetric Key Packages", [RFC 5958](#), DOI 10.17487/RFC5958, August 2010, <<https://www.rfc->

editor.org/info/rfc5958>.

- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in [RFC 2119](#) Key Words", [BCP 14](#), [RFC 8174](#), DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [RFC8208] Turner, S. and O. Borchert, "BGPsec Algorithms, Key Formats, and Signature Formats", [RFC 8208](#), DOI 10.17487/RFC8208, September 2017, <<https://www.rfc-editor.org/info/rfc8208>>.
- [RFC8209] Reynolds, M., Turner, S., and S. Kent, "A Profile for BGPsec Router Certificates, Certificate Revocation Lists, and Certification Requests", [RFC 8209](#), DOI 10.17487/RFC8209, September 2017, <<https://www.rfc-editor.org/info/rfc8209>>.
- [802.1AR] IEEE SA-Standards Board, "IEEE Standard for Local and metropolitan area networks - Secure Device Identity", December 2009, <<http://standards.ieee.org/findstds/standard/802.1AR-2009.html>>.

12.1. Informative References

- [RFC2585] Housley, R. and P. Hoffman, "Internet X.509 Public Key Infrastructure Operational Protocols: FTP and HTTP", [RFC 2585](#), DOI 10.17487/RFC2585, May 1999, <<https://www.rfc-editor.org/info/rfc2585>>.
- [RFC3766] Orman, H. and P. Hoffman, "Determining Strengths For Public Keys Used For Exchanging Symmetric Keys", [BCP 86](#), [RFC 3766](#), DOI 10.17487/RFC3766, April 2004, <<https://www.rfc-editor.org/info/rfc3766>>.
- [RFC5273] Schaad, J. and M. Myers, "Certificate Management over CMS (CMC): Transport Protocols", [RFC 5273](#), DOI 10.17487/RFC5273, June 2008, <<https://www.rfc-editor.org/info/rfc5273>>.
- [RFC5480] Turner, S., Brown, D., Yiu, K., Housley, R., and T. Polk, "Elliptic Curve Cryptography Subject Public Key Information", [RFC 5480](#), DOI 10.17487/RFC5480, March 2009, <<https://www.rfc-editor.org/info/rfc5480>>.
- [RFC5647] Igoe, K. and J. Solinas, "AES Galois Counter Mode for the

Secure Shell Transport Layer Protocol", [RFC 5647](#), DOI 10.17487/RFC5647, August 2009, <<https://www.rfc-editor.org/info/rfc5647>>.

- [RFC5656] Stebila, D. and J. Green, "Elliptic Curve Algorithm Integration in the Secure Shell Transport Layer", [RFC 5656](#), DOI 10.17487/RFC5656, December 2009, <<https://www.rfc-editor.org/info/rfc5656>>.
- [RFC5967] Turner, S., "The application/pkcs10 Media Type", [RFC 5967](#), DOI 10.17487/RFC5967, August 2010, <<https://www.rfc-editor.org/info/rfc5967>>.
- [RFC6187] Igoe, K. and D. Stebila, "X.509v3 Certificates for Secure Shell Authentication", [RFC 6187](#), DOI 10.17487/RFC6187, March 2011, <<https://www.rfc-editor.org/info/rfc6187>>.
- [RFC6470] Bierman, A., "Network Configuration Protocol (NETCONF) Base Notifications", [RFC 6470](#), DOI 10.17487/RFC6470, February 2012, <<https://www.rfc-editor.org/info/rfc6470>>.
- [RFC6484] Kent, S., Kong, D., Seo, K., and R. Watro, "Certificate Policy (CP) for the Resource Public Key Infrastructure (RPKI)", [BCP 173](#), [RFC 6484](#), DOI 10.17487/RFC6484, February 2012, <<https://www.rfc-editor.org/info/rfc6484>>.
- [RFC6668] Bider, D. and M. Baushke, "SHA-2 Data Integrity Verification for the Secure Shell (SSH) Transport Layer Protocol", [RFC 6668](#), DOI 10.17487/RFC6668, July 2012, <<https://www.rfc-editor.org/info/rfc6668>>.
- [RFC7030] Pritikin, M., Ed., Yee, P., Ed., and D. Harkins, Ed., "Enrollment over Secure Transport", [RFC 7030](#), DOI 10.17487/RFC7030, October 2013, <<https://www.rfc-editor.org/info/rfc7030>>.
- [RFC8205] Lepinski, M., Ed., and K. Sriram, Ed., "BGPsec Protocol Specification", [RFC 8205](#), DOI 10.17487/RFC8205, September 2017, <<https://www.rfc-editor.org/info/rfc8205>>.
- [SP800-57] National Institute of Standards and Technology (NIST), Special Publication 800-57: Recommendation for Key Management - Part 1 (Revised), March 2007.

Appendix A. Management/Router Channel Security

Encryption, integrity, authentication, and key exchange algorithms

used by the protected channel SHOULD be of equal or greater strength than the BGPsec keys they protect, which for the algorithm specified in [RFC8208] is 128-bit; see [RFC5480] and by reference [SP800-57] for information about this strength claim as well as [RFC3766] for "how to determine the length of an asymmetric key as a function of a symmetric key strength requirement." In other words, for the encryption algorithm, do not use export grade crypto (40-56 bits of security), do not use Triple DES (112 bits of security). Suggested minimum algorithms would be AES-128: aes128-cbc [RFC4253] and AEAD_AES_128_GCM [RFC5647] for encryption, hmac-sha2-256 [RFC6668] or AESAD_AES_128_GCM [RFC5647] for integrity, ecdsa-sha2-nistp256 [RFC5656] for authentication, and ecdh-sha2-nistp256 [RFC5656] for key exchange.

Some routers support the use of public key certificates and SSH. The certificates used for the SSH session are different than the certificates used for BGPsec. The certificates used with SSH should also enable a level of security commensurate with BGPsec keys; x509v3-ecdsa-sha2-nistp256 [RFC6187] could be used for authentication.

The protected channel must provide confidentiality, authentication, and integrity and replay protection.

Appendix B. The n00b Guide to BGPsec Key Management

This appendix is informative. It attempts to explain all of the PKI technobabble in plainer language.

BGPsec speakers send signed BGPsec updates that are verified by other BGPsec speakers. In PKI parlance, the senders are referred to as signers and the receivers are referred to as relying parties. The signers with which we are concerned here are routers signing BGPsec updates. Signers use private keys to sign and relying parties use the corresponding public keys, in the form of X.509 public key certificates, to verify signatures. The third party involved is the entity that issues the X.509 public key certificate, the Certification Authority (CA). Key management is all about making these key pairs and the certificates, as well as ensuring that the relying parties trust that the certified public keys in fact correspond to the signers' private keys.

The specifics of key management greatly depend on the routers as well as management interfaces provided by the routers' vendor. Because of these differences, it is hard to write a definitive "how to," but this guide is intended to arm operators with enough information to ask the right questions. The other aspect that makes this guide informative is that the steps for the do-it-yourself (DIY) approach

involve arcane commands while the GUI-based vendor-assisted management console approach will likely hide all of those commands behind some button clicks. Regardless, the operator will end up with a BGPsec-enabled router. Initially, we focus on the DIY approach and then follow up with some information about the GUI-based approach.

The first step in the DIY approach is to generate a private key; but in fact what you do is create a key pair; one part, the private key, is kept very private and the other part, the public key, is given out to verify whatever is signed. The two models for how to create the key pair are the subject of this document, but it boils down to either doing it on-router (router-driven) or off-router (operator-driven).

If you are generating keys on the router (router-driven), then you will need to access the router. Again, how you access the router is router-specific, but generally the DIY approach uses the CLI and accessing the router either directly via the router's craft port or over the network on an administrative interface. If accessing the router over the network be sure to do it securely (i.e., use SSHv2). Once logged into the router, issue a command or a series of commands that will generate the key pair for the algorithms referenced in the main body of this document; consult your router's documentation for the specific commands. The key generation process will yield multiple files: the private key and the public key; the file format varies depending on the arcane command you issued, but generally the files are DER or PEM-encoded.

The second step is to generate the certification request, which is often referred to as a certificate signing request (CSR) or PKCS#10 certification request, and to send it to the CA to be signed. To generate the CSR, you issue some more arcane commands while logged into the router; using the private key just generated to sign the certification request with the algorithms referenced in the main body of this document; the CSR is signed to prove to the CA that the router has possession of the private key (i.e., the signature is the proof-of-possession). The output of the command is the CSR file; the file format varies depending on the arcane command you issued, but generally the files are DER or PEM-encoded.

The third step is to retrieve the signed CSR from the router and send it to the CA. But before sending it, you need to also send the CA the subject name (i.e., "ROUTER-" followed by the AS number) and serial number (i.e., the 32-bit BGP Identifier) for the router. The CA needs this information to issue the certificate. How you get the CSR to the CA, is beyond the scope of this document. While you are still connected to the router, install the Trust Anchor (TA) for the root of the PKI. At this point, you no longer need access to the

router for BGPsec-related initiation purposes.

The fourth step is for the CA to issue the certificate based on the CSR you sent; the certificate will include the subject name, serial number, public key, and other fields as well as being signed by the CA. After the CA issues the certificate, the CA returns the certificate, and posts the certificate to the RPKI repository. Check that the certificate corresponds to the private key by verifying the signature on the CSR sent to the CA; this is just a check to make sure that the CA issued a certificate that includes a public key that is the pair of the private key (i.e., the math will work when verifying a signature generated by the private with the returned certificate).

If generating the keys off-router (operator-driven), then the same steps are used as the on-router key generation, (possibly with the same arcane commands as those used in the on-router approach), but no access to the router is needed the first three steps are done on an administrative workstation: o Step 1: Generate key pair; o Step 2: Create CSR and sign CSR with private key, and; o Step 3: Send CSR file with the subject name and serial number to CA.

After the CA has returned the certificate and you have checked the certificate, you need to put the private key and TA in the router. Assuming the DIY approach, you will be using the CLI and accessing the router either directly via the router's craft port or over the network on an admin interface; if accessing the router over the network make doubly sure it is done securely (i.e., use SSHv2) because the private key is being moved over the network. At this point, access to the router is no longer needed for BGPsec-related initiation purposes.

NOTE: Regardless of the approach taken, the first three steps could trivially be collapsed by a vendor-provided script to yield the private key and the signed CSR.

Given a GUI-based vendor-assisted management console, then all of these steps will likely be hidden behind pointing and clicking the way through BGPsec-enabling the router.

The scenarios described above require the operator to access each router, which does not scale well to large networks. An alternative would be to create an image, perform the necessary steps to get the private key and trust anchor on the image, and then install the image via a management protocol.

One final word of advice; certificates include a notAfter field that unsurprisingly indicates when relying parties should no longer trust

the certificate. To avoid having routers with expired certificates follow the recommendations in the Certification Policy (CP) [[RFC6484](#)] and make sure to renew the certificate at least one week prior to the notAfter date. Set a calendar reminder in order not to forget!

Authors' Addresses

Randy Bush
IIJ / Dragon Research Labs
5147 Crystal Springs
Bainbridge Island, Washington 98110
US

Email: randy@psg.com

Sean Turner
sn3rd

Email: sean@sn3rd.com

Keyur Patel
Arrcus, Inc.

Email: keyur@arrcus.com

